


```
PPPPPPPP      AAAAAA      SSSSSSSS      WW      WW      RRRRRRRR      IIIIII      BBBB8888      IIIIII      NN      NN
PPPPPPPP      AAAAAA      SSSSSSSS      WW      WW      RRRRRRRR      IIIIII      BBBB8888      IIIIII      NN      NN
PP      PP      AA      AA      SS      WW      WW      RR      RR      II      BB      BB      II      NN      NN
PP      PP      AA      AA      SS      WW      WW      RR      RR      II      BB      BB      II      NN      NN
PP      PP      AA      AA      SS      WW      WW      RR      RR      II      BB      BB      II      NNNN      NN
PPPPPPPP      AA      AA      SSSSSS      WW      WW      RRRRRRRR      II      BBBB8888      II      NN      NN
PPPPPPPP      AA      AA      SSSSSS      WW      WW      RRRRRRRR      II      BBBB8888      II      NN      NN
PP      AAAAAAAAAA      SS      WW      WW      RR      RR      II      BB      BB      II      NN      NN
PP      AAAAAAAAAA      SS      WW      WW      RR      RR      II      BB      BB      II      NN      NN
PP      AA      AA      SS      WWW      WWW      RR      RR      II      BB      BB      II      NN      NN
PP      AA      AA      SS      WWW      WWW      RR      RR      II      BB      BB      II      NN      NN
PP      AA      AA      SSSSSSSS      WW      WW      RR      RR      IIIIII      BBBB8888      IIIIII      NN      NN
PP      AA      AA      SSSSSSSS      WW      WW      RR      RR      IIIIII      BBBB8888      IIIIII      NN      NN
                                                                ....
                                                                ....
                                                                ....
                                                                ....
```

```
LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
```

.....


```
1 0001 0 MODULE PASSWRITE_BIN ( %TITLE 'Write value in base 2'
2 0002 0 IDENT = '1-002'
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: Pascal Language Support
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains procedures which write a value in
36 0036 1 base 2 to a textfile or string.
37 0037 1
38 0038 1 ENVIRONMENT: User mode - AST reentrant
39 0039 1
40 0040 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. SBL 1-April-1981
45 0045 1 1-002 - Make total-width a longword. SBL 29-June-1982
46 0046 1 --
47 0047 1
```

PASSWRITE_BIN
1-002

Write value in base 2
Declarations

G 11
16-Sep-1984 02:14:25
14-Sep-1984 12:52:01

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASWRIBIN.B32;1

Page 2
(2)

```

: 49      0048 1 %SBTTL 'Declarations'
: 50      0049 1
: 51      0050 1 : PROLOGUE DEFINITIONS:
: 52      0051 1 :
: 53      0052 1
: 54      0053 1 REQUIRE 'RTLIN:PASPROLOG';           ! Externals, linkages, PSECTs, structures
: 55      0117 1
: 56      0118 1
: 57      0119 1 : TABLE OF CONTENTS:
: 58      0120 1 :
: 59      0121 1
: 60      0122 1 FORWARD ROUTINE
: 61      0123 1     PASSWRITE_BIN: NOVALUE,
: 62      0124 1     PASSWRITEV_BIN: NOVALUE;        ! Write to textfile
: 63      0125 1                                     ! Write to string
: 64      0126 1
: 65      0127 1 : MACROS:
: 66      0128 1
: 67      0129 1 :     NONE
: 68      0130 1
: 69      0131 1 : EQUATED SYMBOLS:
: 70      0132 1
: 71      0133 1 :     NONE
: 72      0134 1
: 73      0135 1 : FIELDS:
: 74      0136 1
: 75      0137 1 :     NONE
: 76      0138 1
: 77      0139 1 : OWN STORAGE:
: 78      0140 1
: 79      0141 1 :     NONE
: 80      0142 1
: 81      0143 1
: 82      0144 1 : !! If this is for a V2 system, redefine OTSS$CVT_L_TB as PASS$CVT_L_TB.
: 83      L 0145 1 %IF %VARIANT
: 84      U 0146 1 %THEN
: 85      U 0147 1 UNDECLARE
: 86      U 0148 1     OTSS$CVT_L_TB;
: 87      U 0149 1 EXTERNAL ROUTINE
: 88      U 0150 1     PASS$CVT_L_TB;
: 89      U 0151 1 BIND ROUTINE
: 90      U 0152 1     OTSS$CVT_L_TB = PASS$CVT_L_TB;
: 91      0153 1 %FI
```



```

93 0154 1 %SBTTL 'PASSWRITE_BIN - Write a value in binary'
94 0155 1 GLOBAL ROUTINE PASSWRITE_BIN (
95 0156 1     PFV: REF $PASSPFV_FILE_VARIABLE,
96 0157 1     NBITS,
97 0158 1     VALUE,
98 0159 1     TOTAL_WIDTH: SIGNED,
99 0160 1     ERROR,
100 0161 1     MIN_DIGITS: SIGNED
101 0162 1 ): NOVALUE =
102 0163 1
103 0164 1 ++
104 0165 1 FUNCTIONAL DESCRIPTION:
105 0166 1
106 0167 1     This procedure writes a base 2 representation of a value to the
107 0168 1     specified textfile.
108 0169 1
109 0170 1 CALLING SEQUENCE:
110 0171 1
111 0172 1     CALL PASSWRITE_BIN (PFV.mr.r, NBITS.rl.v, VALUE.rz.r, TOTAL_WIDTH.rl.v
112 0173 1     [, [ERROR.j.r] [, MIN_DIGITS.rl.v ]])
113 0174 1
114 0175 1 FORMAL PARAMETERS:
115 0176 1
116 0177 1     PFV          - The Pascal File Variable (PFV) passed by reference.
117 0178 1                 The structure of the PFV is defined in PASPFV.REQ.
118 0179 1
119 0180 1     NBITS        - The size of VALUE in bits.
120 0181 1
121 0182 1     VALUE        - The address of the value to write.
122 0183 1
123 0184 1     TOTAL_WIDTH  - Total field width.
124 0185 1
125 0186 1     ERROR        - Optional. Address to unwind to if an error occurs.
126 0187 1
127 0188 1     MIN_DIGITS   - Optional. The minimum number of digits to appear
128 0189 1                 in the result. Defaults to the minimum number of
129 0190 1                 digits to represent every bit of the value.
130 0191 1
131 0192 1 IMPLICIT INPUTS:
132 0193 1
133 0194 1     NONE
134 0195 1
135 0196 1 IMPLICIT OUTPUTS:
136 0197 1
137 0198 1     NONE
138 0199 1
139 0200 1 ROUTINE VALUE:
140 0201 1
141 0202 1     NONE
142 0203 1
143 0204 1 SIDE EFFECTS:
144 0205 1
145 0206 1     If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
146 0207 1
147 0208 1 SIGNALLED ERRORS:
148 0209 1
149 0210 1     NEGWIDDIG - negative field width or digits specification is not allowed
```

```
: 150      0211 1 | LINTOOLON - line too long
: 151      0212 1 |
: 152      0213 1 | --
: 153      0214 1 |
: 154      0215 2 BEGIN
: 155      0216 2
: 156      0217 2 LOCAL
: 157      0218 2 FCB: REF $PASS$FCB_CONTROL_BLOCK, | File control block
: 158      0219 2 ACTUAL_DIGITS, | Number of digits actually used
: 159      0220 2 ACTUAL_NBITS, | Value size actually used
: 160      0221 2 DESCR: "BLOCK [8, BYTE], | String descriptor
: 161      0222 2 PFV_ADDR: VOLATILE, | Enable argument
: 162      0223 2 UNWIND_ACT: VOLATILE, | Enable argument
: 163      0224 2 ERROR_ADDR: VOLATILE; | Enable argument
: 164      0225 2
: 165      0226 2 LITERAL
: 166      0227 2 M_SIZE_IN_BITS = %X'04'; | Flags argument for
: 167      0228 2 | convert routine
: 168      0229 2 BUILTIN
: 169      0230 2 ACTUALCOUNT;
: 170      0231 2
: 171      0232 2 ENABLE
: 172      0233 2 PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); | Enable error handler
: 173      0234 2
: 174      0235 2 | +
: 175      0236 2 | Get ERROR parameter, if present.
: 176      0237 2 | -
: 177      0238 2
: 178      0239 2 IF ACTUALCOUNT () GEQU 5
: 179      0240 2 THEN
: 180      0241 2 ERROR_ADDR = .ERROR; | Set unwind address
: 181      0242 2
: 182      0243 2 PFV_ADDR = PFV [PFV$R_PFV]; | Set PFV address
: 183      0244 2
: 184      0245 2 | +
: 185      0246 2 | Validate PFV and get PFV.
: 186      0247 2 | -
: 187      0248 2
: 188      0249 2 PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
: 189      0250 2
: 190      0251 2 | +
: 191      0252 2 | Set unwind action to unlock file.
: 192      0253 2 | -
: 193      0254 2
: 194      0255 2 UNWIND_ACT = PASS$K_UNWIND_UNLOCK;
: 195      0256 2
: 196      0257 2 | +
: 197      0258 2 | Do common initialization.
: 198      0259 2 | -
: 199      0260 2
: 200      0261 2 PASS$INIT_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
: 201      0262 2
: 202      0263 2 | +
: 203      0264 2 | Set initial values for conversion.
: 204      0265 2 | -
: 205      0266 2
: 206      0267 2 ACTUAL_NBITS = .NBITS;
```



```

207 0268 2  ACTUAL_DIGITS = .ACTUAL_NBITS;
208 0269 2
209 0270 2  !+
210 0271 2  ! Create result string descriptor with actual width.
211 0272 2  !-
212 0273 2
213 0274 2  DESCR [DSC$B_CLASS] = DSC$K_CLASS_S;
214 0275 2  DESCR [DSC$B_DTYPE] = DSC$K_DTYPE_T;
215 0276 2  DESCR [DSC$A_POINTER] = .FCB [FCB$A_RECORD_CUR];
216 0277 2  IF ACTUALCOUNT () GEQU 6
217 0278 2  THEN
218 0279 2  BEGIN
219 0280 2  ACTUAL_DIGITS = .MIN_DIGITS;
220 0281 2  IF .ACTUAL_DIGITS LSS 0
221 0282 2  THEN
222 0283 2  $PASSIO_ERROR (PASS$_NEGWIDDIG,0);
223 0284 2  END;
224 0285 2  IF .TOTAL_WIDTH LSS 0
225 0286 2  THEN
226 0287 2  $PASSIO_ERROR (PASS$_NEGWIDDIG,0);
227 0288 2  DESCR [DSC$W_LENGTH] = .TOTAL_WIDTH;
228 0289 2
229 0290 2  !+
230 0291 2  ! Will TOTAL_WIDTH truncate the value?
231 0292 2  !+
232 0293 2  IF .TOTAL_WIDTH LSSU .ACTUAL_NBITS
233 0294 2  THEN
234 0295 2  BEGIN
235 0296 2  !+
236 0297 2  ! Change value size and bit offset to cause a truncated result.
237 0298 2  !-
238 0299 2  ACTUAL_NBITS = .TOTAL_WIDTH;
239 0300 2  END;
240 0301 2
241 0302 2  IF .ACTUAL_DIGITS GTR .TOTAL_WIDTH
242 0303 2  THEN
243 0304 2  ACTUAL_DIGITS = .TOTAL_WIDTH;
244 0305 2
245 0306 2  !+
246 0307 2  ! See if field will fit in record.
247 0308 2  !-
248 0309 2
249 0310 2  BEGIN
250 0311 2  LOCAL
251 0312 2  EXTRA;
252 0313 2  ! Extra characters past end of line
253 0314 2  EXTRA = (.FCB [FCB$A_RECORD_CUR] + .TOTAL_WIDTH) - .FCB [FCB$A_RECORD_END];
254 0315 2  IF .EXTRA GTR 0
255 0316 2  THEN
256 0317 2  $PASSIO_ERROR (PASS$_LINTOOLON,1,.EXTRA);
257 0318 2  END;
258 0319 2
259 0320 2  !+
260 0321 2  ! Do the conversion. It can't fail.
261 0322 2  !-
262 0323 2  OT$SCVT_L_TB (.VALUE, DESCR, .ACTUAL_DIGITS, .ACTUAL_NBITS,
263 0324 2  M_SIZE_IN_BITS);

```

PASSWRITE_BIN
1-002

Write value in base 2
PASSWRITE_BIN - Write a value in binary

K 11
16-Sep-1984 02:14:25
14-Sep-1984 12:52:01

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASWRIBIN.B32;1

Page 6
(3)

```

: 264      0325 2      FCB [FCB$A_RECORD_CUR] = .FCB [FCB$A_RECORD_CUR] + .DESCR [DSC$W_LENGTH];
: 265      0326 2
: 266      0327 2
: 267      0328 2
: 268      0329 2      !+ Call WRITE epilogue routine to move the last character written to the
: 269      0330 2      !- user's buffer and to unlock the file variable.
: 270      0331 2
: 271      0332 2
: 272      0333 2      PASS$END_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
: 273      0334 2
: 274      0335 2      RETURN;
: 275      0336 2
: 276      0337 1      END;
```

! End of routine PASSWRITE_BIN

.TITLE PASSWRITE_BIN Write value in base 2
.IDENT \1-002\

.EXTRN PASSWRITE_BIN, PASSWRITEV_BIN
.EXTRN PASS\$IO_HANDLER
.EXTRN PASS\$VALIDATE_PFV
.EXTRN PASS\$INIT_WRITE
.EXTRN PASS\$SIGNAL, PASS\$K_NEGWIDDIG
.EXTRN PASS\$K_LINTOOLON
.EXTRN OTSS\$CVT_L_TB, PASS\$END_WRITE

.PSECT _PASS\$CODE, NOWRT, SHR, PIC, 2

			01FC 00000	.ENTRY PASSWRITE_BIN, Save R2,R3,R4,R5,R6,R7,R8	: 0155
58	00000000G	00	9E 00002	MOVAB PASS\$SIGNAL, R8	
5E		10	C2 00009	SUBL2 #16, SP	
		7E	D4 0000C	CLRL ERROR_ADDR	: 0215
	04	AE	7C 0000E	CLRQ UNWIND_ACT	
6D	009A	CF	DE 00011	MOVAL 8\$, (FP)	
05		6C	91 00016	CMPB (AP), #5	: 0239
		04	1F 00019	BLSSU 1\$	
6E	14	AC	DO 0001B	MOVL ERROR, ERROR_ADDR	: 0241
56	04	AC	DO 0001F	MOVL PFV, R6	: 0243
08	AE	56	DO 00023	MOVL R6, PFV_ADDR	
	00000000G	00	16 00027	JSB PASS\$VALIDATE_PFV	: 0249
04	AE	01	DO 0002D	MOVL #1, UNWIND_ACT	: 0255
	00000000G	00	16 00031	JSB PASS\$INIT_WRITE	: 0261
54	08	AC	DO 00037	MOVL NBITS, ACTUAL_NBITS	: 0267
53		54	DO 0003B	MOVL ACTUAL_NBITS, ACTUAL_DIGITS	: 0268
0E	AE	010E	8F B0 0003E	MOVW #270, DESCR+2	: 0275
10	AE	EC	A7 DO 00044	MOVL -20(FCB), DESCR+4	: 0276
	06	6C	91 00049	CMPB (AP), #6	: 0277
		06	1F 0004C	BLSSU 2\$	
53	18	AC	DO 0004E	MOVL MIN_DIGITS, ACTUAL_DIGITS	: 0280
		06	19 00052	BLSS 3\$	: 0281
52	10	AC	DO 00054	MOVL TOTAL_WIDTH, R2	: 0285
		0A	18 00058	BGEQ 4\$	
		7E	D4 0005A	CLRL -(SP)	: 0287
7E	00G	8F	9A 0005C	MOVZBL #PASS\$K_NEGWIDDIG, -(SP)	
68		02	FB 00060	CALLS #2, PASS\$SIGNAL	
		04	00063	RET	
0C	AE	52	B0 00064	MOVW R2, DESCR	: 0288

PASSWRITE_BIN
1-002

Write value in base 2
PASSWRITE_BIN - Write a value in binary

L 11
16-Sep-1984 02:14:25
14-Sep-1984 12:52:01

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASWRIBIN.B32;1

Page 7
(3)

54		52	D1	00068	CMPL	R2, ACTUAL_NBITS	: 0293
		03	1E	00068	BGEQU	5\$	: 0299
54		52	D0	0006D	MOVL	R2, ACTUAL_NBITS	: 0302
52		53	D1	00070	CMPL	ACTUAL_DIGITS, R2	: 0304
		03	15	00073	BLEQ	6\$	: 0313
53		52	D0	00075	MOVL	R2, ACTUAL_DIGITS	: 0314
52	EC	A7	C1	00078	ADDL3	-20(FCB), R2, R0	: 0316
50	FO	A7	C2	0007D	SUBL2	-16(FCB), EXTRA	: 0323
		0C	15	00081	BLEQ	7\$	: 0326
		50	DD	00083	PUSHL	EXTRA	: 0333
		01	DD	00085	PUSHL	#1	: 0337
7E	00G	8F	9A	00087	MOVZBL	#PASSK_LINTOOLON, -(SP)	: 0215
68		03	FB	0008B	CALLS	#3, PASS\$\$SIGNAL	: 0333
		04	0008E	RET			: 0337
		04	DD	0008F	PUSHL	#4	: 0215
		18	BB	00091	PUSHR	#^M<R3,R4>	: 0333
	18	AE	9F	00093	PUSHAB	DESCR	: 0337
	0C	AC	DD	00096	PUSHL	VALUE	: 0215
00000000G	00	05	FB	00099	CALLS	#5, OTS\$CVT_L_TB	: 0333
	50	OC	AE	3C	MOVZWL	DESCR, R0	: 0337
EC	A7	50	CO	000A4	ADDL2	R0, -20(FCB)	: 0215
		00	16	000A8	JSB	PASS\$END_WRITE	: 0333
		04	000AE	RET			: 0337
		0000	000AF	8\$:	.WORD	Save nothing	: 0215
50	08	AC	D0	000B1	MOVL	8(AP), R0	: 0333
50	04	A0	D0	000B5	MOVL	4(R0), R0	: 0337
	EC	A0	9F	000B9	PUSHAB	ERROR_ADDR	: 0215
	FO	A0	9F	000BC	PUSHAB	UNWIND_ACT	: 0333
	F4	A0	9F	000BF	PUSHAB	PFV_ADDR	: 0337
		03	DD	000C2	PUSHL	#3	: 0215
		5E	DD	000C4	PUSHL	SP	: 0333
00000000G	7E	04	AC	7D	MOVQ	4(AP), -(SP)	: 0337
	00	03	FB	000CA	CALLS	#3, PASS\$\$IO_HANDLER	: 0215
		04	000D1	RET			: 0333

; Routine Size: 210 bytes, Routine Base: _PASS\$CODE + 0000

: 277 0338 1
: 278 0339 1 !<BLF/PAGE>

PASSWRITE_BIN
1-002

Write value in base 2

PASSWRITEV_BIN - Write value in base 2 to string

M 11

16-Sep-1984 02:14:25

14-Sep-1984 12:52:01

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASWRIBIN.B32;1

Page 8
(4)

```

280 0340 1 %SBTTL 'PASSWRITEV_BIN - Write value in base 2 to string'
281 0341 1 GLOBAL ROUTINE PASSWRITEV_BIN (
282 0342 1     MAX_LENGTH: WORD,           ! Maximum length of string
283 0343 1     STRING_LINE: REF VECTOR [, WORD], ! String to write to
284 0344 1     NBITS,                   ! Number of bits in VALUE
285 0345 1     VALUE: REF VECTOR [, BYTE], ! Value to write
286 0346 1     TOTAL_WIDTH: SIGNED,      ! Total field width
287 0347 1     ERROR,                   ! Error unwind address
288 0348 1     MIN_DIGITS: SIGNED        ! Minimum number of digits
289 0349 1 ) : NOVALUE =
290 0350 1
291 0351 1 ++
292 0352 1 FUNCTIONAL DESCRIPTION:
293 0353 1
294 0354 1     This procedure writes a value in base 2 to the specified string.
295 0355 1
296 0356 1 CALLING SEQUENCE:
297 0357 1
298 0358 1     CALL PASSWRITEV_BIN (MAX_LENGTH.rw.v, STRING_LINE.wvt.r,
299 0359 1     NBITS.rl.v, VALUE.rz.v, TOTAL_WIDTH.rl.v
300 0360 1     [, [ERROR.j.r] [, MIN_DIGITS.fl.v]])
301 0361 1
302 0362 1 FORMAL PARAMETERS:
303 0363 1
304 0364 1     MAX_LENGTH      - The maximum length of STRING_LINE.
305 0365 1
306 0366 1     STRING_LINE    - A varying string to which the output will be appended.
307 0367 1
308 0368 1     NBITS           - Size of VALUE in bits.
309 0369 1
310 0370 1     VALUE          - The value to write.
311 0371 1
312 0372 1     TOTAL_WIDTH    - The width of the field to write.
313 0373 1
314 0374 1     ERROR           - Optional. If specified, the address to unwind to
315 0375 1     in case of an error.
316 0376 1
317 0377 1     MIN_DIGITS      - Minimum number of digits to write. Defaults to
318 0378 1     the minimum necessary to represent every bit of
319 0379 1     the value.
320 0380 1
321 0381 1 IMPLICIT INPUTS:
322 0382 1
323 0383 1     NONE
324 0384 1
325 0385 1 IMPLICIT OUTPUTS:
326 0386 1
327 0387 1     NONE
328 0388 1
329 0389 1 ROUTINE VALUE:
330 0390 1
331 0391 1     NONE
332 0392 1
333 0393 1 SIDE EFFECTS:
334 0394 1
335 0395 1     NONE
336 0396 1
```


PASSWRITE_BIN
1-002

Write value in base 2
PASSWRITEV_BIN - Write value in base 2 to strin

N 11
16-Sep-1984 02:14:25
14-Sep-1984 12:52:01

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASWRIBIN.B32;1

Page 9
(4)

```

337 0397 1 | SIGNALLED ERRORS:
338 0398 1 |
339 0399 1 |     See PASSWRITE_BIN
340 0400 1 |
341 0401 1 | --
342 0402 1 |
343 0403 2 | BEGIN
344 0404 2 |
345 0405 2 | LOCAL
346 0406 2 |     PFV: $PASSPFV_FILE_VARIABLE,      ! Pascal File Variable
347 0407 2 |     ARG_LIST: VECTOR [7, LONG],       ! Argument list
348 0408 2 |     PFV_ADDR: VOLATILE,               ! Enable argument
349 0409 2 |     UNWIND_ACT: VOLATILE,             ! Enable argument
350 0410 2 |     ERROR_ADDR: VOLATILE;             ! Enable argument
351 0411 2 |
352 0412 2 | BUILTIN
353 0413 2 |     ACTUALCOUNT;                    ! Count of arguments
354 0414 2 |
355 0415 2 | ENABLE
356 0416 2 |     PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler
357 0417 2 |
358 0418 2 | !+
359 0419 2 | ! Get ERROR parameter, if present.
360 0420 2 | !-
361 0421 2 |
362 0422 2 | IF ACTUALCOUNT () GEQU 6
363 0423 2 | THEN
364 0424 2 |     ERROR_ADDR = .ERROR;              ! Set unwind address
365 0425 2 |
366 0426 2 | PFV_ADDR = PFV [PFV$R_PFV];           ! Set PFV address
367 0427 2 |
368 0428 2 | !+
369 0429 2 | ! Set up ARG_LIST.
370 0430 2 | !-
371 0431 2 |
372 0432 2 | ARG_LIST [0] = 4;                     ! Four arguments
373 0433 2 | ARG_LIST [1] = PFV [PFV$R_PFV];       ! PFV address
374 0434 2 | ARG_LIST [2] = .NBITS;                 ! Number of bits
375 0435 2 | ARG_LIST [3] = VALUE [0];             ! Value to write
376 0436 2 | ARG_LIST [4] = .TOTAL_WIDTH;          ! Field width
377 0437 2 | IF ACTUALCOUNT () GEQU 7
378 0438 2 | THEN
379 0439 2 | BEGIN
380 0440 2 |     ARG_LIST [0] = 6;                 ! Add two more arguments
381 0441 2 |     ARG_LIST [5] = 0;                 ! Error address
382 0442 2 |     ARG_LIST [6] = .MIN_DIGITS;       ! Minimum digits
383 0443 2 | END;
384 0444 2 |
385 0445 2 | !+
386 0446 2 | ! Call PASS$DO_WRITEV to do the work, giving it the address of
387 0447 2 | ! PASSWRITE_BIN to call.
388 0448 2 | !-
389 0449 2 |
390 0450 2 | PASS$DO_WRITEV (PFV [PFV$R_PFV], .MAX_LENGTH, STRING_LINE [0], ARG_LIST,
391 0451 2 |     PASSWRITE_BIN);
392 0452 2 |
393 0453 2 | RETURN;
```


PASS\$WRITE_BIN
1-002

Write value in base 2

PASS\$WRITEV_BIN - Write value in base 2 to strin

B 12
16-Sep-1984 02:14:25
14-Sep-1984 12:52:01

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASWRIBIN.B32;1

Page 10
(4)

: 394
: 395

0454 2
0455 1

END;

! End of routine PASS\$WRITEV_BIN

```

                                007C 00000
                                34 C2 00002
                                7E D4 00005
                                04 AE 7C 00007
                                6D 004F CF DE 0000A
                                06 6C 91 0000F
                                04 1F 00012
                                6E 18 AC D0 00014
                                08 AE 28 AE 9E 00018 1$:
                                0C AE 04 D0 0001D
                                10 AE 28 AE 9E 00021
                                14 AE 0C AC 7D 00026
                                1C AE 14 AC D0 0002B
                                07 6C 91 00030
                                0C 1F 00033
                                0C AE 06 D0 00035
                                24 AE 20 AE D4 00039
                                55 FEE9 CF 9E 00041 2$:
                                54 0C AE 9E 00046
                                56 28 AE 9E 0004A
                                53 08 AC D0 0004E
                                52 04 AC 3C 00052
                                00000000G 00 16 00056
                                04 0005C
                                0000 0005D 3$:
                                50 08 AC D0 0005F
                                50 04 A0 D0 00063
                                C8 A0 9F 00067
                                CC A0 9F 0006A
                                D0 A0 9F 0006D
                                03 DD 00070
                                5E DD 00072
                                7E 04 AC 7D 00074
                                00000000G 00 03 FB 00078
                                04 0007F
```

.EXTRN PASS\$DO_WRITEV

```

.ENTRY PASS$WRITEV_BIN, Save R2,R3,R4,R5,R6
SUBL2 #52, SP
CLRL ERROR_ADDR
CLRQ UNWIND_ACT
MOVAL 3$, (FP)
CMPB (AP), #6
BLSSU 1$
MOVL ERROR, ERROR_ADDR
MOVAB PFV, PFV_ADDR
MOVL #4, ARG_LIST
MOVAB PFV, ARG_LIST+4
MOVQ NBIT5, ARG_LIST+8
MOVL TOTAL_WIDTH, ARG_LIST+16
CMPB (AP), #7
BLSSU 2$
MOVL #6, ARG_LIST
CLRL ARG_LIST+20
MOVL MIN_DIGITS, ARG_LIST+24
MOVAB PASS$WRITE_BIN, R5
MOVAB ARG_LIST, R4
MOVAB PFV, R6
MOVL STRING_LINE, R3
MOVZWL MAX_LENGTH, R2
JSB PASS$DO_WRITEV
RET
.WORD Save nothing
MOVL 8(AP), R0
MOVL 4(R0), R0
PUSHAB ERROR_ADDR
PUSHAB UNWIND_ACT
PUSHAB PFV_ADDR
PUSHL #3
PUSHL SP
MOVQ 4(AP), -(SP)
CALLS #3, PASS$IO_HANDLER
RET
```

: 0341
: 0403
: 0422
: 0424
: 0426
: 0432
: 0433
: 0434
: 0436
: 0437
: 0440
: 0441
: 0442
: 0450
: 0455
: 0403
:

; Routine Size: 128 bytes, Routine Base: _PASS\$CODE + 00D2

: 396
: 397

0456 1
0457 1 !<BLF/PAGE>

PAS\$WRITE_BIN
1-002

Write value in base 2
PAS\$WRITEV_BIN - Write value in base 2 to strin

C 12
16-Sep-1984 02:14:25
14-Sep-1984 12:52:01

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASWRIBIN.B32;1

Page 11
(5)

: 399
: 400
: 401

0458 1 END
0459 1
0460 0 ELUDOM

! End of module PAS\$WRITE_BIN

:
:
:
:

PSECT SUMMARY

:
:
:

NameBytesAttributes

:_PAS\$CODE338 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

:
:
:
:

Library Statistics

:
:
:

FileTotalSymbolsLoadedPercentPagesMappedProcessingTime

_\$255\$DUA28:[SYSLIB]STARLET.L32;197766058100:01.0

_\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;142797223300:00.4

:
:
:

COMMAND QUALIFIERS

:
:
:

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:PASWRIBIN/OBJ=OBJ\$:PASWRIBIN MSRC\$:PASWRIBIN/UPDATE=(ENH\$:PASWRIBIN)

:
:
:
:
:
:
:
:

Size: 338 code + 0 data bytes

: Run Time: 00:08.9

: Elapsed Time: 00:29.3

: Lines/CPU Min: 3094

: Lexemes/CPU-Min: 14825

: Memory Used: 101 pages

: Compilation Complete

0297

AH-BT13A-SE
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY